ESD-TDR-64-583

AD 609524

47-1

COPY 2 OF 3 96

HARD COPY $. 2.00

MICROFICHE $. 0.50

Group Report 1964-64

Algorithms for Estimating A Re-entry Body's Position, Velocity and Ballistic Coefficient in Real Time or for Post Flight Analysis

F. C. Schweppe

3 December 1964

Prepared under Electronic Systems Division Contract AF 19(628)-500 by

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Lexington, Massachusetts

This Document Contains
Missing Page/s That Are
Unavailable In The
Original Document

OR are
Blank pgs
that have
Been Removed

**BEST
AVAILABLE COPY**

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, with the support of the U.S. Air Force under Contract AF 19(628)-500.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LINCOLN LABORATORY

ALGORITHMS FOR ESTIMATING A RE-ENTRY BODY'S POSITION, VELOCITY AND BALLISTIC COEFFICIENT IN REAL TIME OR FOR POST FLIGHT ANALYSIS

F. C. SCHWEPPE

Group 28

GROUP REPORT 1964-64

3 DECEMBER 1964

LEXINGTON MASSACHUSETTS

ABSTRACT

A general theory of recursive estimation is applied to the particular problem of estimating the time history of a re-entry vehicle's position, velocity and ballistic coefficient from radar measurements of range, elevation, azimuth and range rate. The resulting algorithms have the following properties:

1) Conceptually simple and easy to implement on either general purpose or special purpose computers
2) Noniterative
3) Storage requirements independent of amount of data
4) Extremely fast; capable of real time operation
5) A finite memory span; that is, can act like a "sliding arc" algorithm.

Explicit formulae are provided along with general discussions on the basic concept.

Accepted for the Air Force
Stanley J. Wisniewski
Lt Colonel, USAF
Chief, Lincoln Laboratory Office

ALGORITHMS FOR ESTIMATING A RE-ENTRY BODY'S POSITION, VELOCITY AND BALLISTIC COEFFICIENT IN REAL TIME OR FOR POST FLIGHT ANALYSIS

1. INTRODUCTION

This report presents explicit computational algorithms for reducing radar observations of range, azimuth, elevation and range rate to an estimate of the time behavior of a re-entry vehicle's position, velocity and ballistic coefficient (or weight-to-drag ratio, β). The algorithms are conceptually simple and computationally very fast. Computer storage requirements are small and independent of the amount of data processed. Iterative techniques are not used. The algorithms can employ a finite effective memory span. Thus knowledge of the exact equations of motion and an assumption of constant β are not required as slow variations in β can be tracked. The algorithms are for real time data processing. Their high speed also makes them desirable for post-flight analysis when large amounts of data are to be processed. Implemented on appropriate computer hardware, they enable fruitful man-machine interaction.

The algorithms are recursive in nature and the basic concept is not new. Reference 1 states it; Refs. 2 and 3 analyze its behavior for certain situations; Ref. 4 contains a tutorial discussion on the linear theory which underlies it; and these references are only samples; see for example, Refs. 5 and 6. The basic concept is presently being used for estimating position and velocity from range, elevation and azimuth observations in the real time system for the Tradex radar on Kwajalein.* The Kwajalein algorithm employs a useful type of approximation which we also discuss. The purpose of the present report is simply the collation of the existing material into a form more directly suitable for implementation. Hopefully our presentation combines formula and concepts almost ready for the programmer's pencil with generalities

* This application of the basic concept was developed by Ken Ralston of Lincoln Lab. independently from the studies reported in Refs. 1 - 4. His work provided the motivation for this report.

sufficient for understanding both the concept's power and its shortcomings.

Notational requirements cause many of our equations to appear rather foreboding. Therefore we preview the simplicity of the resulting algorithms by giving Fig. 1.1, which is a verbal flow diagram for one of our algorithms. (The same diagram expressed in mathematical symbols is given in Sec. 5, Fig. 5.1.) The weighting matrices are defined by recursive equations capable of real time numerical solution. In a special case of practical importance, explicit, closed form expressions for the weighting matrices are given.

An obvious and important question is; how does the accuracy of a fast, simple scheme such as Fig. 1.1 compare with say an iterative solution of the Maximum Likelihood equations? In this report, we do not address this question as it requires the fairly advanced mathematics used in Refs. 2 and 3. However some answers are known. Under certain conditions which often occur in practice, our algorithms are as efficient as Maximum Likelihood. These conditions are listed in Sec. 3. Under more general conditions (albeit Gaussian errors), the iterative Maximum Likelihood approach intuitively seems to be better, but mathematical proofs of such intuition are not available.

In Sec. 2 we go through some preliminaries and in Sec. 3 present the two basic algorithms. Approximations for a special case are considered in Sec. 4. Various computational aspects are covered in Sec. 5 with a final discussion given in Sec. 6.

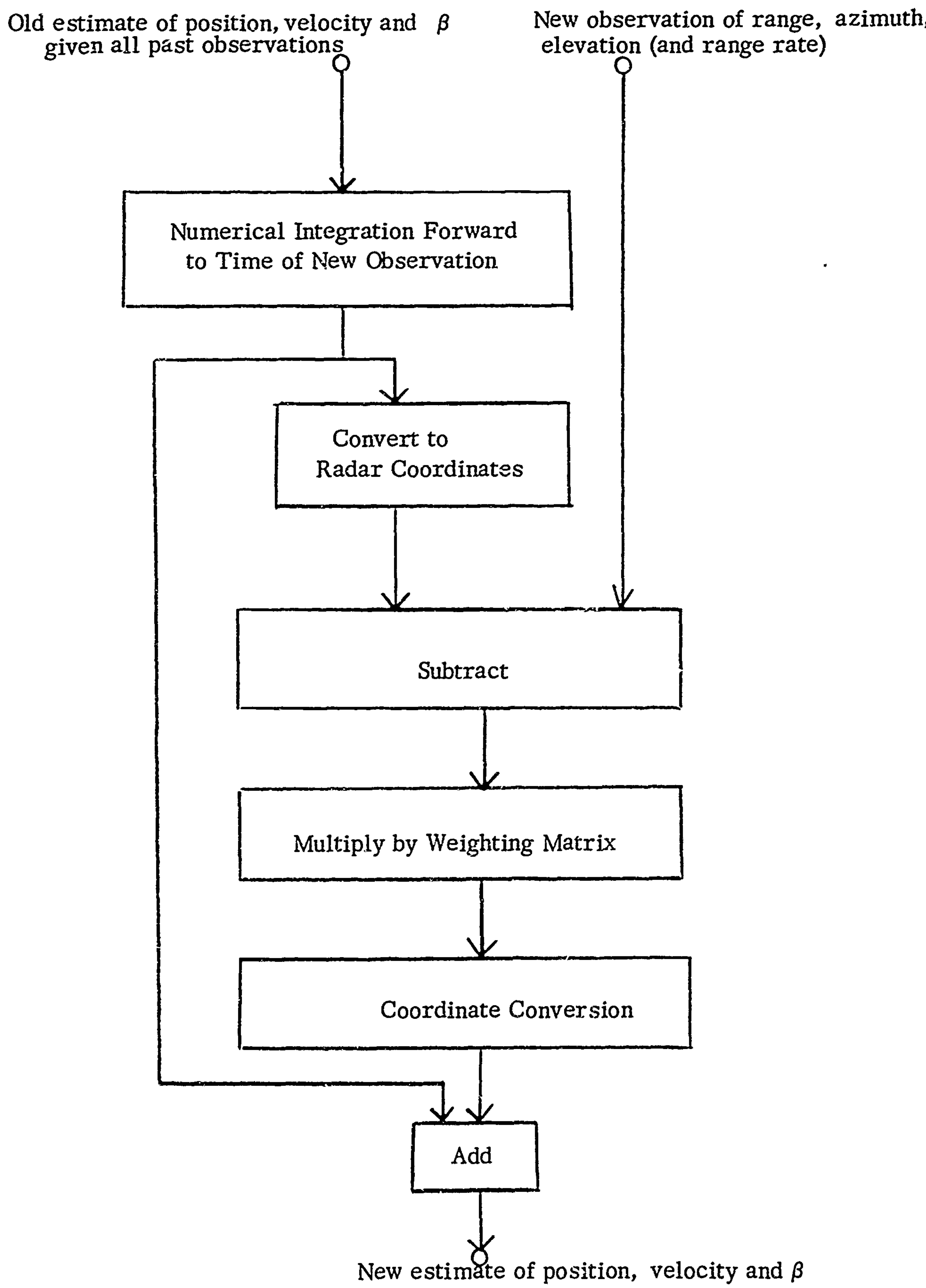

Fig. 1.1 Flow diagram for one algorithm (see also Fig. 5.1).

2. PRELIMINARIES

We cryptically present some definitions and formulas which will be used in the later developments.

2.1. Basic Model

For the present, we assume the re-entry body is moving in a known force field on a trajectory that is completely specified by a parameter vector consisting of seven elements; the six components of position and velocity at some instant of time and β, the weight to drag ratio. Let the vector $\underline{p}(t)$

$$\underline{p}(t) = \begin{bmatrix} p_1(t) \\ p_2(t) \\ p_3(t) \\ p_4(t) \\ p_5(t) \\ p_6(t) \end{bmatrix}$$

denote the body's position and velocity in a cartesian, inertial, earth centered coordinate system where $p_k(t)$, $k = 1, 2, 3$, are the position components and $p_k(t)$, $k = 4, 5, 6$, are the velocity components. The time behavior of this vector is governed by the vector system of differential equations

$$\frac{d}{dt}\, \underline{p}(t) = \underline{f}[\underline{p}(t),\ t] \tag{2.1}$$

where the elements of the vector $\underline{f}$ are given by

$$f_k[\underline{p}(t), t] = p_{k+3}(t) \quad k = 1, 2, 3$$

$$f_k[\underline{p}(t), t] = \frac{-\mu p_{k-3}(t)}{[p_1^2(t)+p_2^2(t)+p_3^2(t)]^{3/2}} + \frac{d_k(t)}{\beta} \quad k = 4, 5, 6 \tag{2.2}$$

where μ is the gravitational constant and $d_k(t)/\beta$ is the acceleration due to drag given by

$$\frac{d_k(t)}{\beta} = \frac{\tilde{p}_k(t)[\tilde{p}_4^2(t) + \tilde{p}_5^2(t) + \tilde{p}_6^2(t)]^{1/2}\,\rho(t)}{2W/C_d A}$$

where $\beta = W/C_d A$, $\rho(t)$ is the atmospheric density and $\tilde{p}_k(t)$, $k = 4, 5, 6$ are the inertial velocities corrected to account for the atmospheric rotation.

Define the object's position and velocity at time t relative to a radar on the earth's surface by:

$r(t)$	range
$\theta(t)$	azimuth
$\phi(t)$	elevation
$\dot{r}(t)$	range rate
$\dot{\theta}(t)$	azimuth rate
$\dot{\phi}(t)$	elevation rate.

Further, define the seven dimensional vector $\underline{w}(t)$ by

$$
\underline{w}(t) = \begin{bmatrix} r(t) \\ \theta(t) \\ \phi(t) \\ \dot{r}(t) \\ \dot{\theta}(t) \\ \dot{\phi}(t) \\ \alpha(t) \end{bmatrix} = \begin{bmatrix} w_1(t) \\ w_2(t) \\ w_3(t) \\ w_4(t) \\ w_5(t) \\ w_6(t) \\ w_7(t) \end{bmatrix}
$$

where

$$
\alpha(t) = \frac{\rho(t)}{\beta} .
$$

Knowledge of $\underline{w}(t)$ at $t = t_n$ combined with Eq. (2.1) enables the calculation of $\underline{w}(t_{n+1})$ for any value of t_{n+1}. Define

$$
\underline{w}(t_{n+1}) = \Psi[\underline{w}(t_n), t_{n+1}, t_n] \tag{2.3}
$$

as the corresponding function. The evaluation of Eq. (2.3) can be done by:

1) Conversion of $r(t_n), \theta(t_n), \phi(t_n), \dot{r}(t_n), \dot{\theta}(t_n), \dot{\phi}(t_n)$ to inertial coordinates $\underline{p}(t_n)$
2) Numerical integration of Eq. (2.1) from t_n to t_{n+1} to get $\underline{p}(t_{n+1})$
3) Conversion of $\underline{p}(t_{n+1})$ back to radar coordinates
4) Evaluation of $\rho(t_{n+1}) - \rho(t_n)$ from a model atmosphere.

The reasons for our choice of the coordinate system $\underline{w}(t)$ and in particular, the definition of $\alpha(t)$ will become evident later on. We discuss in Sec. 5 the use of other coordinate systems and methods of calculating Eq. (2.3).

Define $\Theta[t_n, t_m, \underline{w}^o(t_m)]$ as the 7 by 7 matrix formed from the partial derivatives of the elements of $\underline{w}(t_n)$ with respect to the elements of $\underline{w}(t_m)$, evaluated for $\underline{w}(t_m) =$

$\underline{w}^o(t_m)$. Thus,

$$\Theta[t_n, t_m, \underline{w}^o(t_m)] = \left. \begin{bmatrix} \frac{\partial w_1(t_n)}{\partial w_1(t_m)} & \cdots & \frac{\partial w_1(t_n)}{\partial w_7(t_m)} \\ \vdots & & \vdots \\ \frac{\partial w_7(t_n)}{\partial w_1(t_m)} & & \frac{\partial w_7(t_n)}{\partial w_7(t_m)} \end{bmatrix} \right/ _{\underline{w}(t_m) = \underline{w}^o(t_m)} . \tag{2.4}$$

Let the q-dimensional vector $\underline{y}(t_n)$ denote the observation made by the radar at time t_n. Assume

$$\underline{y}(t_n) = H\underline{w}(t_n) + \underline{v}(t_n) \tag{2.5}$$

where H is a q×7 matrix and $\underline{v}(t_n)$, $n = 1, 2, \ldots,$ is a sequence of zero mean vector Gaussian random variables with

$$E[\underline{v}(t_n)\underline{v}'(t_m)] = \begin{cases} 0 & n \neq m \\ Q(t_n) & n = m \end{cases}$$

where the prime denotes transpose. To illustrate the use of this model consider the case where at times t_n, $n = 1, 2, \ldots,$ the radar measures range, elevation, azimuth and range rate with stationary, independent, errors of variations $\sigma_r^2, \sigma_\theta^2, \sigma_\phi^2, \sigma_{\dot{r}}^2$, respectively. Then

$$q = 4$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{2.6}$$

$$Q(t_n) = Q = \begin{bmatrix} \sigma_r^2 & 0 & 0 & 0 \\ 0 & \sigma_\theta^2 & 0 & 0 \\ 0 & 0 & \sigma_\phi^2 & 0 \\ 0 & 0 & 0 & \sigma_{\dot{r}}^2 \end{bmatrix} . \tag{2.7}$$

Correlation between the errors (say between range and range rate errors) are handled by "filling in" $Q(t_n)$.

Define

$\hat{\underline{w}}(t_n/m_2, m_1)$: The estimate of $\underline{w}(t_n)$ made from $\underline{y}(t_k)$, $k=m_1, \ldots, m_2$

$\hat{\underline{w}}(t_n/m)$: The estimate of $\underline{w}(t_n)$ made from $\underline{y}(t_k)$, $k=1, \ldots, m$.

The purpose of this report is the presentation of computational algorithms for $\hat{\underline{w}}(t_n/m_2, m_1)$ and $\hat{\underline{w}}(t_n/m)$. An estimate of $\alpha(t_n)$ of course provides an estimate of β and we often refer to $\hat{\alpha}(t_n)$ as an estimate of β.

2.2. Estimation for Linear Dynamical System

We now change the subject and consider some results from the theory of parameter estimation for linear dynamical systems. Consider the system

$$\underline{x}(t_n) = \Phi(t_n, t_{n-1})\underline{x}(t_{n-1}) \tag{2.8}$$

where Φ is a square, invertable matrix. Assume we make the observation $\underline{z}(t_n)$ of the form

$$\underline{z}(t_n) = H\underline{x}(t_n) + \underline{v}(t_n)$$

where H and $\underline{v}(t_n)$ are as defined and discussed with respect to Eq. (2.5). Define $\hat{\underline{x}}(t_n/m_2, m_1)$ and $\hat{\underline{x}}(t_n/m)$ in direct analogy with the $\hat{\underline{w}}(t_n/m_2, m_1)$ and $\hat{\underline{w}}(t_n/m)$.

For the growing memory case we have the following formula:*

$$\hat{\underline{x}}(t_n/n) = \hat{\underline{x}}(t_n/n-1) + \mathcal{I}^{-1}(n/n)H'Q^{-1}(t_n)[\underline{z}(t_n) - H\hat{\underline{x}}(t_n/n-1)] \tag{2.9}$$

$$\begin{aligned}\hat{\underline{x}}(t_n/n-1) &= \Phi(t_n, t_{n-1})\hat{\underline{x}}(t_{n-1}/n-1) \\ \mathcal{I}(n/n) &= \Phi^{-1'}(t_n, t_{n-1})\mathcal{I}(n-1/n-1)\Phi^{-1}(t_n, t_{n-1}) \\ &\quad + H'Q^{-1}(t_n)H\end{aligned} \tag{2.10}$$

where "-1" denotes matrix inversion.

For the finite memory case we have the following formula:†

$$\begin{aligned}\hat{\underline{x}}(t_n/n, n-\tau) &= \hat{\underline{x}}(t_n/n-1, n-\tau-1) + \\ &\mathcal{I}^{-1}(n/n, n-\tau)\{H'Q^{-1}(t_n)[\underline{z}(t_n) - H\hat{\underline{x}}(t_n/n-1, n-\tau-1] - \\ &\Phi^{-1'}(t_n, t_{n-\tau-1})H'Q^{-1}(t_{n-\tau-1})[\underline{z}(t_{n-\tau-1}) - \\ &H\hat{\underline{x}}(t_{n-\tau-1}/n-1, n-\tau-1)]\}\end{aligned} \tag{2.11}$$

* These are explicitly given in Section 3.3 of Ref. 4.

† These are not explicitly derived in Ref. 4 but the reference contains the ideas necessary for their derivation.

where

$$\mathcal{I}(n/n, n-\tau) = \Phi^{-1'}(t_n, t_{n-1})\,\mathcal{I}(n-1/n-1, n-\tau-1)\,\Phi^{-1}(t_n, t_{n-1}) + H'Q^{-1}(t_n)H - \Phi^{-1'}(t_n, t_{n-\tau-1})H'Q^{-1}(t_{n-\tau-1})H\Phi^{-1}(t_n, t_{n-\tau-1}) \tag{2.12}$$

$$\hat{\underline{x}}(t_n/n-1, n-\tau-1) = \Phi(t_n, t_{n-1})\hat{\underline{x}}(t_{n-1}/n-1, n-\tau-1)$$

$$\hat{\underline{x}}(t_{n-\tau-1}/n-1, n-\tau-1) = \Phi^{-1}(t_{n-1}, t_{n-\tau-1})\hat{\underline{x}}(t_{n-1}/n-1, n-\tau-1) \quad .$$

The preceding formulae appear complex but they actually have a simple structure which we now unveil in hopes of providing the reader with some insight.* Consider first the growing memory case of Eqs. (2.9) and (2.10). Define

$\mathcal{I}(n/m)$:	The information matrix which measures the amount of information on $\underline{x}(t_n)$ contained in $\underline{z}(t_k)$, $k = 1, \ldots, m$.
$\mathcal{I}(n)$:	The information matrix which measures the amount of information on $\underline{x}(t_n)$ contained in $\underline{z}(t_n)$.

$$\mathcal{I}(n) = H'Q^{-1}(t_n)H$$

$$\mathcal{I}(n+1/m) = \Phi^{-1'}(t_{n+1}, t_n)\,\mathcal{I}(n/m)\Phi^{-1}(t_{n+1}, t_n) \quad .$$

With these definitions, Eq. (2.10) is simply a statement that information matrices are additive; i.e., Eq. (2.10) can be rewritten as

* Equations (2.9) through (2.12) are written in a form which leads most readily to the nonlinear problem of real interest.

$$\mathcal{I}(n/n) = \mathcal{I}(n/n-1)+\mathcal{I}(n) \ .$$

Now define

$\underline{\mathcal{B}}(n/m)$:	The information vector which is the actual information on $\underline{x}(t_n)$ contained in $\underline{z}(t_k)$, $k = 1, \ldots, m$.
$\underline{\mathcal{B}}(n)$:	The information vector which is the actual information on $\underline{x}(t_n)$ contained in $\underline{z}(t_n)$.

$$\underline{\mathcal{B}}(n) = H'Q^{-1}(t_n)\underline{z}(t_n)$$

$$\underline{\mathcal{B}}(n/m) = \mathcal{I}(n/m)\hat{\underline{x}}(n/m)$$

$$\underline{\mathcal{B}}(n/m) = \Phi^{-1'}(t_n, t_{n-1}) \ (n-1/m)$$

$$\hat{\underline{x}}(n/m) = \Phi(t_n, t_{n-1})\hat{\underline{x}}(n-1/m) \quad .$$

With these definitions, algebraic manipulation shows that Eq. (2.9) is simply a statement that the information vectors themselves are also additive; that is, Eq. (2.9) can be rewritten as

$$\underline{\mathcal{B}}(n/n) = \underline{\mathcal{B}}(n/n-1) + \underline{\mathcal{B}}(n) \ .$$

The same type of interpretation can be given to the finite memory case of Eqs. (2.11) and (2.12). Define

$\mathcal{I}(n/m_2, m_1)$:	The information matrix for $z(t_k)$, $k = m_1, \ldots, m_2$.

Note that $\mathcal{I}(n/m, m)$ is the amount of information on $\underline{x}(t_n)$ contained in $\underline{z}(t_m)$ and thus

$$\mathcal{I}(n/n, n) = \mathcal{I}(n) \quad .$$

Then Eq. (2.12) can be rewritten as

$$\begin{aligned} \mathcal{I}(n/n, n-\tau) &= \mathcal{I}(n/n-1, n-1-\tau) \\ &\quad + \mathcal{I}(n) - \mathcal{I}(n/n-\tau-1, n-\tau-1) \end{aligned}$$

that is, Eq. (2.12) simply states that we add the information matrix for $\underline{z}(t_n)$ and subtract the information matrix for $\underline{z}(t_{n-\tau-1})$. Define

$\underline{\mathcal{B}}(n/m_2, m_1)$: The information vector for $\underline{z}(t_k)$, $k=m_1, \ldots, m_2$.

Note

$$\underline{\mathcal{B}}(n/n, n) = \underline{\mathcal{B}}(n) \quad .$$

Then by algebraic manipulation, Eq. (2.11) can be rewritten as:

$$\begin{aligned} \underline{\mathcal{B}}(n/n, n-\tau) &= \underline{\mathcal{B}}(n/n-1, n-\tau-1) \\ &\quad + \underline{\mathcal{B}}(n) - \underline{\mathcal{B}}(n/n-\tau-1, n-\tau-1) \end{aligned}$$

which says we simply add and subtract the appropriate information vectors.

The terms, information matrix and information vector, are used as the corresponding concepts can be directly related to information theory.* Reference 7

* Our information matrix is often called the Fisher Information Matrix. The term, information vector, is not standard jargon. It is closely related to the concept of a sufficient statistic.

contains complete mathematical discussions on this subject while the Appendix of Ref. 8 contains a simplified discussion.

The information matrix has the following very useful physical interpretation,

$$\mathcal{I}^{-1}(n/m) = E\{[\underline{\hat{x}}(t_n/m) - \underline{x}(t_n)][\underline{\hat{x}}(t_n/m) - \underline{x}(t_n)]'\} \tag{2.13}$$

that is, $\mathcal{I}^{-1}(n/m)$ is the covariance matrix of the errors in $\underline{\hat{x}}(t_n/m)$. Reference 4 discusses why the algorithms of Eqs. (2.9) and (2.10) give the minimum possible $\mathcal{I}^{-1}(n/m)$ under the constraint

$$E[\underline{\hat{x}}(t_n/m)] = \underline{x}(t_n) .$$

An analogous interpretation can also be given $\mathcal{I}^{-1}(n/m_2, m_1)$.

3. THE ALGORITHMS

We now present the algorithms of interest. We do not derive them in any mathematical sense, but the motivation behind their choice is straightforward. Consider Eq. (2.3). A Taylor series expansion about some estimate $\hat{\underline{w}}(t_{n-1})$ gives

$$\begin{aligned}\underline{w}(t_n) &= \Psi[\hat{\underline{w}}(t_{n-1}),\ t_n, t_{n-1}] \\ &\quad + \Theta[t_n, t_{n-1}, \hat{\underline{w}}(t_{n-1})]\ [\underline{w}(t_{n-1}) - \hat{\underline{w}}(t_{n-1})] \qquad (3.1) \\ &\quad + \text{Remainder terms}\end{aligned}$$

where Θ is the matrix of partial derivatives of Eq. (2.4). Now assume $\underline{w}(t_{n-1}) - \hat{\underline{w}}(t_{n-1})$ is small enough to make the remainder terms negligible. It is then reasonable to use the linear data processing algorithms of Sec. 2.2 where we make the following associations:

$$\underline{x}(t_n) \sim \underline{w}(t_n) - \Psi[\hat{\underline{w}}(t_{n-1}), t_n, t_{n-1}]$$

$$\underline{z}(t_n) \sim \underline{y}(t_n) - H\Psi[\hat{\underline{w}}(t_{n-1}), t_n, t_{n-1}]$$

$$\Phi(t_n, t_{n-1}) \sim \Theta[t_n, t_{n-1}, \hat{\underline{w}}(t_{n-1})] \ .$$

In Sec. 2.1, we assumed that β is constant and that the equations of motion of the re-entry body are as given by Eq. (2.3). During re-entry, this is often a naive assumption. However, over short periods of time, β is essentially constant* and the body's motion is well approximated by Eq. (2.3). Therefore we handle the real problem of interest by limiting the memory span of the algorithms. This gives a tracking action whereby we can follow slow variations, in say β.

* For the present algorithms, we do not attempt to estimate variations in drag due to body oscillations. Thus, with respect to any such oscillatory effects, we estimate an average β.

One basic algorithm is as follows:

$$\hat{\underline{w}}(t_n/n) = \hat{\underline{w}}(t_n/n-1) + \mathcal{I}^{-1}(n/n)\, H' Q^{-1}(t_n)\, [\underline{y}(t_n) - H\, \hat{\underline{w}}(t_n/n-1)] \tag{3.2}$$

$$\hat{\underline{w}}(t_n/n-1) = \Psi[\hat{\underline{w}}(t_{n-1}/n-1), t_n, t_{n-1}] \tag{3.3}$$

$$\begin{aligned} \mathcal{I}(n/n) = {} & \Theta^{-1'}[t_n, t_{n-1}, \hat{\underline{w}}(t_{n-1}/n-1)]\, \mathcal{I}(n-1/n-1)\, \Theta^{-1}[t_n, t_{n-1}, \hat{\underline{w}}(t_{n-1}/n-1)] \\ & + H' Q^{-1}(t_n) H \qquad \text{for } n \le n_o \\ = {} & \mathcal{I}(n_o/n_o) \qquad \text{for } n > n_o \ . \end{aligned} \tag{3.4}$$

This algorithm is of course analogous to the growing memory formulae of Eqs. (2.9) and (2.10). The main difference is the "clamping" of $\mathcal{I}(n/n)$ at a fixed value, $\mathcal{I}(n_o/n_o)$ for $n > n_o$. If $\mathcal{I}(n/n)$ is not clamped, it goes to zero and $\hat{\underline{x}}(t_n/n)$ converges to a single unique trajectory as n increases. This would be desired if β were constant and Eq. (2.3) were exact.* The clamping action limits the effective memory span of the algorithm and thereby provides the desired tracking ability.

The second basic algorithm is directly analogous to Eqs. (2.11) and (2.12) and thus achieves a finite memory span without subterfuge.

$$\begin{aligned} & \hat{\underline{w}}(t_n/n, n-\tau) = \hat{\underline{w}}(t_n/n-1, n-\tau-1) + \\ & \mathcal{I}^{-1}(n/n, n-\tau)\{H' Q^{-1}(t_n)[\underline{y}(t_n) - H\hat{\underline{w}}(t_n/n-1, n-\tau-1)] - \\ & \Theta'^{-1}(t_n, t_{n-\tau-1})\, H' Q^{-1}(t_{n-\tau-1})[\underline{y}(t_{n-\tau-1}) - H\hat{\underline{w}}(t_{n-\tau-1}/n-1, n-\tau-1)]\} \end{aligned} \tag{3.5}$$

* Reference 2 analyzes the behavior of $\hat{\underline{w}}(t_n/n)$ under these conditions for the scalar case and gives conditions which guarantee convergence and asymptotic efficiency.

$$I(n/n, n-\tau) = \Theta'^{-1}(t_n, t_{n-1}) I(n-1/n-1, n-\tau-1)\, \Theta^{-1}(t_n, t_{n-1}) +$$
$$H'Q^{-1}(t_n)H - \Theta'^{-1}(t_n, t_{n-\tau-1}) H'Q^{-1}(t_{n-\tau-1}) H\, \Theta^{-1}(t_n, t_{n-\tau-1}) \tag{3.6}$$

$$\underline{\hat{w}}(t_n/n-1, n-\tau-1) = \Psi[\underline{\hat{w}}(t_{n-1}/n-1, n-\tau-1), t_n, t_{n-1}] \tag{3.7}$$

$$\underline{\hat{w}}(t_{n-\tau-1}/n-1, n-\tau-1) = \Psi[\underline{\hat{w}}(t_{n-1}/n-1, n-\tau-1), t_{n-\tau-1}, t_{n-1}] \tag{3.8}$$

where in Eqs. (3.5) and (3.6) we have simplified the notation by writing

$$\Theta[t_n, t_m] = \Theta[t_n, t_m, \underline{\hat{w}}(t_m/m, m-\tau)] \ .$$

Equation (3.8) requires the numerical integration of Eq. (2.1) from t_{n-1} backwards in time to $t_{n-\tau-1}$. This may be acceptable but if not, the following formulae can be used instead:

$$\underline{\hat{w}}(t_{n-\tau-1}/n-1, n-\tau-1) = \underline{\hat{w}}(t_{n-\tau-1}/n-2, n-\tau-2)$$
$$+ \ \Theta^{-1}(t_{n-1}, t_{n-\tau-1}) \, [\underline{\hat{w}}(t_{n-1}/n-1, n-\tau-1) - \underline{\hat{w}}(t_{n-1}/n-2, n-\tau-2)] \ . \tag{3.9}$$

The algorithm requires storage of the observations, $\underline{y}(t_k)$; $k = n, \ldots, n-\tau-1$ (and the $\underline{\hat{w}}(t_{k-\tau-1}/k-1, k-\tau-1)$ if Eq. (3.9) is used). However for most cases of interest, this requirement does not limit the technique's usefulness.

As stated, both algorithms evaluate Ψ each time a new vector observation is obtained; that is, we relinearize as per Eq. (3.1) with each new observation. However, in many high data rate radars, relinearization only every r^{th} vector observation is

required. In such cases we can effectively combine the r, q-dimensional observations into a single 7-dimensional observation and the algorithm corresponding to Eqs. (3.2) and (3.4) is

$$\underline{\hat{w}}(t_n/n) = \underline{\hat{w}}(t_n/n-r) + \mathcal{I}^{-1}(n/n) \sum_{k=n-r+1}^{n} \{\Theta'^{-1}(t_n, t_k) H' Q^{-1}(t_k)[\underline{y}(t_k) - H\underline{\hat{w}}(t_k/n-r)]\} \tag{3.10}$$

$$\mathcal{I}(n/n) = \Theta'^{-1}(t_n, t_{n-r}) \mathcal{I}(n-r/n-r) \Theta^{-1}(t_n, t_{n-r}) + \sum_{k=n-r+1}^{n} \Theta'^{-1}(t_n, t_k) H' Q^{-1}(t_k) H \Theta^{-1}(t_n, t_k) \quad . \tag{3.11}$$

where we have again simplified the Θ notation. Equations (3.5) through (3.9) for the finite memory case can be modified in a similar fashion. This reduction in the number of relinearizations can save computer time as will be discussed in Sec. 5.

Both algorithms are recursive in nature. Initial conditions for the finite memory algorithm of Eqs. (3.5) through (3.9) can be obtained from the growing memory formulae for $n = \tau$; that is, we can use Eqs. (3.2) through (3.4) (for $n_0 \geq \tau$) to start up the finite memory scheme. The initial conditions required for Eqs. (3.2) and (3.4) are $\underline{\hat{w}}(t_1/0)$ and $\mathcal{I}(1/0)$; that is, an *apriori* guess on the value $\underline{w}(t_1)$ and a measure of how good the *apriori* guess is. [$\mathcal{I}^{-1}(1/0)$ can be considered the covariance matrix of the errors in the *apriori* guess of $\underline{w}(t_1)$ as per the discussion of Eq. (2.13).] If $\mathcal{I}(1/0) = 0$, then the *apriori* guess is assumed to have "infinite errors" associated with it and it will not be weighed into any future estimates. However, if $\mathcal{I}(1/0) = 0$, $\mathcal{I}^{-1}(n/n)$ will not exist until the total number of observations, nq, (q is dimension of the observation vector) equals or exceeds 7. There are various ways to handle this problem such as

1) Make $I(1/0)$ small but not zero

2) Combine the first few observations into a single vector observations as per the discussions of the preceding paragraph.

The choice depends on the particular problem at hand.

In Sec. 2.2 we interpreted $I^{-1}(n/n)$ and $I^{-1}(n/n, n-\tau)$ as the covariance matrices of the errors in the estimates, $\hat{\underline{x}}(t_n/n)$ and $\hat{\underline{x}}(t_n/n, n-\tau)$. The errors in the estimates $\hat{\underline{w}}(t_n/n, n-\tau)$ can be similarly interpreted in terms of $I^{-1}(n/n, n-\tau)$ of Eq. (3.6) provided:

1) The estimate errors are small enough so that the remainder terms of Eq. (3.1) are truly negligible and

2) β is "nearly" constant and Eq. (2.3) is a "good" approximation over the memory span.

The relationship between the errors in $\hat{\underline{w}}(t_n/n)$ for $n \geq n_o$ and $I^{-1}(n_o/n_o)$ have not yet been investigated but under the above two conditions, they should be "in the same general ball park."

The two conditions of the preceding paragraph are, of course, the conditions which lead to the choice of the algorithms themselves. When the conditions are not satisfied, it is very difficult to perform explicit error analyses but the algorithm performance should degrade gracefully as the conditions are violated. If the algorithm fails, it will probably be due to one or a combination of the following factors:

1) An incomplete error model; i.e., the neglect of error sources such as bias errors or correlations

2) Insufficient data; i.e., good estimates are impossible using just the observations made over the memory span

3) Bad initial conditions.

The recursive scheme can be modified to handle a more extensive error model. The

problem of insufficient data can be solved by buying a more expensive radar. If the choice of initial conditions, $\hat{\underline{w}}(t_1/0)$, is poor, the algorithm may search around for a while until it locks on to the trajectory. If $\hat{\underline{w}}(t_1/0)$ is very bad, it may fail to lock on at all. For post flight analyses this is no problem as different initial conditions can be tried.* For real time applications, failure to lock on must be avoided and, if necessary, an auxialiary algorithm used to provide satisfactory initial conditions.

Our use of terms such as "transient," "search," "lock on" and "track" is deliberate as the algorithms are directly analogous to a feedback control system (or servomechanism) designed to track an input signal and its time derivatives. This analogy is mentioned as the author has found it to be a valuable aid.

* For example, an initial transient can be removed by feedback of the $\hat{\underline{w}}(t_n/n)$ obtained after lock on.

4. A SPECIAL CASE

In Sec. 3 we gave the algorithms in a general form. We now restrict interest to a special case and introduce some approximations of the type found in the Kwajalein algorithm. This enables certain closed form expressions which greatly reduce computation time and complexity. In addition, some of the behavioral properties of the algorithms can be displayed.

Assume we observe range, elevation, azimuth and range rate and that the errors in these observations are independent of each other and are stationary. Then we have H and $Q(t_n) = Q$ as given by Eq. (2.6) and Eq. (2.7). The case where the range rate observation is not present is included as a special case. Assume further that the observations are equally spaced in time by Δ units of time. Thus

$$t_n = t_{n-k} + \Delta k \quad .$$

The crucial approximation is to assume $\Theta[t_n, t_{n-k}, \hat{\underline{w}}(t_{n-k})]$ is of the form

$$\Theta[t_n, t_{n-k}, \hat{\underline{w}}(t_{n-k})] = \begin{bmatrix} 1 & 0 & 0 & \Delta k & 0 & 0 & \frac{(\Delta k)^2 \hat{\gamma}_r}{2} \\ 0 & 1 & 0 & 0 & \Delta k & 0 & \frac{(\Delta k)^2 \hat{\gamma}_\theta}{2} \\ 0 & 0 & 1 & 0 & 0 & \Delta k & \frac{(\Delta k)^2 \hat{\gamma}_\phi}{2} \\ 0 & 0 & 0 & 1 & 0 & 0 & \Delta k \hat{\gamma}_r \\ 0 & 0 & 0 & 0 & 1 & 0 & \Delta k \hat{\gamma}_\theta \\ 0 & 0 & 0 & 0 & 0 & 1 & \Delta k \hat{\gamma}_\phi \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.1}$$

where

$$\hat{\gamma}_r = \hat{v}(t_{n-k})\hat{\dot{r}}(t_{n-k})$$

$$\hat{\gamma}_\theta = \hat{v}(t_{n-k})\hat{\dot{\theta}}(t_{n-k})\cos[\hat{\phi}(t_{n-k})]$$

$$\hat{\gamma}_\phi = \hat{v}(t_{n-k})\hat{\dot{\phi}}(t_{n-k})$$

$$\hat{v}^2(t_{n-k}) = \hat{\dot{r}}^2(t_{n-k}) + \hat{r}^2(t_{n-k})\{\hat{\dot{\phi}}^2(t_{n-k}) + \hat{\dot{\theta}}^2(t_{n-k})\cos^2[\hat{\phi}(t_{n-k})]\} \ .$$

It is easily seen that the $\hat{\gamma}_r, \hat{\gamma}_\theta$ and $\hat{\gamma}_\phi$ are simply the coefficients that split the total acceleration due to drag into its respective, r, θ and ϕ coordinates.

The motivation behind the choice of Eq. (4.1) is as follows. Consider a small variation, $\delta \underline{w}(t+\Delta k)$, in $\underline{w}(t+\Delta k)$ due to small variations, $\delta \underline{w}(t)$, in $\underline{w}(t)$. Then to a first approximation

$$\delta \underline{w}(t+\Delta k) = \Theta(t+\Delta k, t)\,\delta \underline{w}(t)$$

where we have again simplified the Θ notation. Now consider the range coordinate. Equation (4.1) states,

$$\delta r(t+\Delta k) = \delta r(t) + \Delta k\, \delta \dot{r}(t) + \frac{(\Delta k)^2}{2}\gamma_r \delta\alpha(t)$$

that is we have made the approximations

$$\begin{aligned} \frac{\partial r(t+\Delta k)}{\partial r(t)} &= 1 \\ \frac{\partial r(t+\Delta k)}{\partial \dot{r}(t)} &= \Delta k \\ \frac{\partial r(t+\Delta k)}{\partial \alpha(t)} &= \frac{(\Delta k)^2}{2}\hat{\gamma}_r \end{aligned} \tag{4.2}$$

and in addition, have neglected the effect of the perturbations, $\delta\theta(t)$, $\delta\dot{\theta}(t)$, $\delta\phi(t)$ and $\delta\dot{\phi}(t)$. These perturbation effects are dropped because the corresponding partial derivatives are small; a valid step provided the corresponding perturbations, $\delta\theta(t)$, .. are not too large. The incorporation of $(\Delta k)^2$ behavior only with respect to $\delta\alpha(t)$, is motivated by the assumption that in practice, $\delta\alpha(t)$ will be relatively large. Equation (4.1) models variations in $\alpha(t_n)$ as constant. Section 5 discusses the use of exponential atmospheres and Eq. (5.2) can be used to check this approximation.

The $\hat{\gamma}_r, \hat{\gamma}_\theta$ and $\hat{\gamma}_\phi$ are time varying quantities but in many applications, they will be essentially constant over the memory span of the algorithms. Thus we make the additional assumption that they are constants, denoted by $\gamma_r, \gamma_\theta, \gamma_\phi$, and let $\Theta(t_n, t_{n-k})$ denote the corresponding version of Eq. (4.1). With this final approximation, it is possible to obtain closed form analytic expressions for $I^{-1}(n/n)$ and $I^{-1}(n/n, n-\tau)$.

Under our various assumptions,

$$I(n/n, n-\tau+1) = I(\tau/\tau) \quad .$$

Thus we need consider only $I^{-1}(n/n)$. Equation (3.4) can be written as (for $n \leq n_0$)

$$I(n/n) = \sum_{k=0}^{n-1} \Theta^{-1\prime}(t_n, t_{n-k}) H' Q^{-1} H \Theta^{-1}(t_n, t_{n-k}) \quad . \tag{4.3}$$

Evaluation of $I^{-1}(n/n)$ proceeds easiest by use of the orthonormal polynomials defined by

$$\lambda_0(k\Delta) = \frac{1}{\sqrt{n}}$$

$$\lambda_1(k\Delta) = \frac{1}{\Delta}\sqrt{\frac{12}{n(n^2-1)}}\left[k\Delta - \frac{(n-1)\Delta}{2}\right]$$

$$\lambda_2(k\Delta) = \frac{1}{\Delta^2}\sqrt{\frac{180}{n(n^2-1)(n^2-4)}}\left[(k\Delta)^2 - (n-1)k\Delta^2 + \Delta^2\,\frac{(n-1)(n-2)}{6}\right]$$

where

$$\sum_{k=0}^{n-1} \lambda_i(k\Delta)\,\lambda_j(k\Delta) = \begin{cases} 1 & i=j \\ 0 & i\neq j \end{cases} \qquad (4.4)$$

These orthonormal polynomials are discussed in many places, see, for example, Refs. 9 and 10 (which are orthonormal over $k=1,\ldots,n$). Define

$$\Lambda(k\Delta) = \left[\begin{array}{cccccc:c}
\sqrt{n}\,\lambda_o(k\Delta) & 0 & 0 & -\Delta\sqrt{\frac{n(n^2-1)}{12}}\,\lambda_1(k\Delta) & 0 & 0 & \frac{\gamma_r\Delta^2}{2}\sqrt{\frac{n(n^2-1)(n^2-4)}{180}}\,\lambda_2(k\Delta) \\
0 & \sqrt{n}\,\lambda_o(k\Delta) & 0 & 0 & -\Delta\sqrt{\frac{n(n^2-1)}{12}}\,\lambda_1(k\Delta) & 0 & \frac{\gamma_\theta\Delta^2}{2}\sqrt{\frac{n(n^2-1)(n^2-4)}{180}}\,\lambda_2(k\Delta) \\
0 & 0 & \sqrt{n}\,\lambda_o(k\Delta) & 0 & 0 & -\Delta\sqrt{\frac{n(n^2-1)}{12}}\,\lambda_1(k\Delta) & \frac{\gamma_\phi\Delta^2}{2}\sqrt{\frac{n(n^2-1)(n^2-4)}{180}}\,\lambda_2(k\Delta) \\
0 & 0 & 0 & \sqrt{n}\,\lambda_o(k\Delta) & 0 & 0 & -\gamma_r\Delta\sqrt{\frac{n(n^2-1)}{12}}\,\lambda_1(k\Delta) \\
0 & 0 & 0 & 0 & \sqrt{n}\,\lambda_o(k\Delta) & 0 & -\gamma_r\Delta\sqrt{\frac{n(n^2-1)}{12}}\,\lambda_1(k\Delta) \\
0 & 0 & 0 & 0 & 0 & \sqrt{n}\,\lambda_o(k\Delta) & -\gamma_\phi\Delta\sqrt{\frac{n(n^2-1)}{12}}\,\lambda_1(k\Delta) \\
0 & 0 & 0 & 0 & 0 & 0 & \sqrt{n}\,\lambda_o(k\Delta)
\end{array}\right]$$

Eq. (4.5)

$$
D = \begin{bmatrix}
1 & 0 & 0 & \frac{(n-1)\Delta}{2} & 0 & 0 & \frac{\gamma_r \Delta^2 (n-1)(n-2)}{12} \\
0 & 1 & 0 & 0 & \frac{(n-1)\Delta}{2} & 0 & \frac{\gamma_\theta \Delta^2 (n-1)(n-2)}{12} \\
0 & 0 & 1 & 0 & 0 & \frac{(n-1)\Delta}{2} & \frac{\gamma_\phi \Delta^2 (n-1)(n-2)}{12} \\
0 & 0 & 0 & 1 & 0 & 0 & \frac{\gamma_r (n-1)\Delta}{2} \\
0 & 0 & 0 & 0 & 1 & 0 & \frac{\gamma_\theta (n-1)\Delta}{2} \\
0 & 0 & 0 & 0 & 0 & 1 & \frac{\gamma_\phi (n-1)\Delta}{2} \\
0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} .
$$

Eq. (4.6)

Then

$$\Theta^{-1}(t_n, t_{n-k}) D = \Lambda(k\Delta) \ . \tag{4.7}$$

Substitution of Eq. (4.7) into (4.3) gives

$$\mathcal{I}^{-1}(n/n) = D \mathcal{I}_\Lambda^{-1} D' \tag{4.8}$$

where

$$\mathcal{I}_\Lambda = \sum_{k=0}^{n-1} \Lambda'(k\Delta) H' Q^{-1} H \Lambda(k\Delta) \quad .$$

Equations (2.6) and (2.7) for H and Q, and the orthonormal properties of the $\lambda_j(k\Delta)$. Eq. (4.4), result in I_Λ being a diagonal matrix with main diagonal elements, I_{jj}, given by

$$
\begin{aligned}
I_{11} &= \frac{n}{\sigma_r^2} \\
I_{22} &= \frac{n}{\sigma_\theta^2} \\
I_{33} &= \frac{n}{\sigma_\phi^2} \\
I_{44} &= \frac{\Delta^2 n(n^2-1)}{12\sigma_r^2} + \frac{n}{\sigma_{\dot r}^2} \\
I_{55} &= \frac{\Delta^2 n(n^2-1)}{12\sigma_\theta^2} \\
I_{66} &= \frac{\Delta^2 n(n^2-1)}{12\sigma_\phi^2} \\
I_{77} &= \frac{\xi^2\Delta^4 n(n^2-1)(n^2-4)}{720} + \frac{\gamma_r^2\Delta^2 n(n^2-1)}{\sigma_{\dot r}^2\, 12}
\end{aligned}
\tag{4.9}
$$

where

$$\xi^2 = \frac{\gamma_r^2}{\sigma_r^2} + \frac{\gamma_\theta^2}{\sigma_\theta^2} + \frac{\gamma_\phi^2}{\sigma_\phi^2} \quad . \tag{4.10}$$

Since the inversion of diagonal matrices is considered to be trivial, the calculation of the final closed form expression for $I^{-1}(n/n)$ now proceeds directly by doing the matrix multiplication of Eq. (4.8) using Eqs. (4.6) and (4.9). In order to display the

result in a relatively simple form, we partition $\underline{I}^{-1}(n/n)$ as

$$\underline{I}^{-1}(n/n) = \begin{bmatrix} A_{pp} & A_{pv} & A_{p\beta} \\ A'_{pv} & A_{vv} & A_{v\beta} \\ A'_{p\beta} & A'_{v\beta} & A_{\beta\beta} \end{bmatrix} \tag{4.11}$$

where

A_{pp}	3×3 matrix corresponding to the position coordinates
A_{vv}	3×3 matrix corresponding to the velocity coordinates
$A_{\beta\beta}$	1×1 matrix corresponding to the $\alpha(t)$ coordinate

and $A_{pv}, A_{p\beta}, A_{vp}$ are the resulting off-diagonal matrices. Define

$$\Gamma = \begin{bmatrix} \gamma_r \\ \gamma_\theta \\ \gamma_\phi \end{bmatrix} .$$

Then

$$A_{pp} = \begin{bmatrix} \frac{\sigma_r^2}{n} + \frac{(n-1)^2\Delta^2}{4\underline{I}_{44}} & 0 & 0 \\ 0 & \frac{\sigma^2(4_{n-2})}{n(n+1)} & 0 \\ 0 & 0 & \frac{\sigma^2(4_{n-2})}{n(n+1)} \end{bmatrix} + \frac{\Delta^4(n-1)^2(n-2)^2}{144\underline{I}_{77}} \Gamma\Gamma'$$

$$A_{pv} = \begin{bmatrix} \frac{(n-1)\Delta}{2I_{44}} & 0 & 0 \\ 0 & \frac{6\sigma_\theta^2}{\Delta n(n+1)} & 0 \\ 0 & 0 & \frac{6\sigma_\theta^2}{\Delta n(n+1)} \end{bmatrix} + \frac{\Delta^3(n-1)^2(n-2)}{24I_{77}}\,\Gamma\Gamma'$$

$$A_{p\beta} = \frac{\Delta^2(n-1)(n-2)}{12I_{77}}\,\Gamma$$

$$A_{vv} = \begin{bmatrix} \frac{1}{I_{44}} & 0 & 0 \\ 0 & \frac{12\sigma_\theta^2}{\Delta^2 n(n-1)^2} & 0 \\ 0 & 0 & \frac{12\sigma_\theta^2}{\Delta^2 n(n-1)^2} \end{bmatrix} + \frac{(n-1)^2\Delta^2}{4I_{77}}\,\Gamma\Gamma'$$

$$A_{v\beta} = \frac{(n-1)\Delta}{2I_{77}}\,\Gamma$$

$$A_{\beta\beta} = \frac{1}{I_{77}} \quad .$$

Note that Eqs. (2.6) and (2.7) give

$$H'Q^{-1} = \begin{bmatrix} \frac{1}{\sigma_r^2} & 0 & 0 & 0 \\ 0 & \frac{1}{\sigma_\theta^2} & 0 & 0 \\ 0 & 0 & \frac{1}{\sigma_\phi^2} & 0 \\ 0 & 0 & 0 & \frac{1}{\sigma_{\dot{r}}^2} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} .$$

Thus the 4×7 matrix $I^{-1}(n/n)H'Q^{-1}$ can be calculated by inspection from Eq. (4.11). This 4×7 matrix can then be substituted directly in Eq. (3.2) to completely specify that algorithm. The necessary expressions for the algorithm of Eq. (3.5) are obtained in a similarly easy fashion.

The case where only range, azimuth and elevation are observed can be obtained from our expressions for $\tau^{-1}(n/n)$ simply by letting $\sigma_{\dot{r}}^2 \to \infty$. When this is done, $\tau^{-1}(n/n)$ develops many more symmetries.

The approximation of Eq. (4.1) and the constancy of the γ_r, γ_θ and γ_ϕ are valid assumptions over only short periods of time. Thus if their consequences are to be used, the effective memory span (τ or n_0) of the algorithm cannot be allowed to become too large even if β is constant and the equations of motion are known over longer periods of time.

The approximation of Eq. (4.1) is also useful in the insight it provides on the performance of the algorithms. Assume, as per the discussions of Sec. 3, that conditions are such that $I^{-1}(n/n)$ represents the covariance matrix of the errors. An instructive case occurs when doppler is not present and the re-entry body's velocity vector lies entirely in the radial direction to give $\gamma_\theta = \gamma_\phi = 0$. For simplicity we also assume $n \gg 1$. Then Eq. (4.11) reduces to

$$I^{-1}(n/n) = \begin{bmatrix} \frac{9\sigma_r^2}{n} & 0 & 0 & \frac{36\sigma_r^2}{\Delta n^2} & 0 & 0 & \frac{60\sigma_r^2}{\gamma_r \Delta^2 n^3} \\ 0 & \frac{4\sigma_\theta^2}{n} & 0 & 0 & \frac{6\sigma_\theta^2}{\Delta n^2} & 0 & 0 \\ 0 & 0 & \frac{4\sigma_\phi^2}{n} & 0 & 0 & \frac{6\sigma_\phi^2}{\Delta n^2} & 0 \\ \frac{36\sigma_r^2}{\Delta n^2} & 0 & 0 & \frac{192\sigma_r^2}{\Delta^2 n^3} & 0 & 0 & \frac{360\sigma_r^2}{\gamma_r \Delta^3 n^4} \\ 0 & \frac{6\sigma_\theta^2}{\Delta n^2} & 0 & 0 & \frac{12\sigma_\theta^2}{\Delta^2 n^3} & 0 & 0 \\ 0 & 0 & \frac{6\sigma_\phi^2}{\Delta n^2} & 0 & 0 & \frac{12\sigma_\phi^2}{\Delta^2 n^3} & 0 \\ \frac{60\sigma_r^2}{\gamma_r \Delta^2 n^3} & 0 & 0 & \frac{360\sigma_r^2}{\gamma_r \Delta^3 n^4} & 0 & 0 & \frac{720\sigma_r^2}{\gamma_r^2 \Delta^4 n^5} \end{bmatrix}$$

Eq. (4.12)

An important question is the performance of our algorithms if we use them outside the atmosphere where we actually should not try to estimate β. Such a procedure would alleviate the need for an algorithm change as the re-entry occurs. The resulting degradation can be measured by repeating our analyses for the case where β need not be estimated. The result is as follows; if we use our algorithms exo-atmospherically, we unnecessarily degrade the range accuracy by $\sqrt{9/4} \approx 1.5$ and the range-rate accuracy by $\sqrt{192/12} = 4$. The amount of degradation can be somewhat misleading as we are estimating the range and range rate at the time of the last observation and at this point, the range-range rate estimate errors are highly correlated. The picture changes if we compare the errors in the estimates of range and range rate as calculated at the midpoint of the memory span; that is, if for the finite memory algorithm, we consider the errors in $\hat{r}(t_m/n, n-\tau)$ and $\hat{\dot{r}}(t_m/n, n-\tau)$ where $t_m = t_{n-\tau} + \Delta\tau/2$. This is the case of interest in post flight analysis. At midpoint, the range-range rate estimate errors are uncorrelated; the degradation in range is still $\sqrt{9/4} \approx 1.5$; but there is no degradation in range rate accuracy. These results are taken from Ref. 9 which contains curves showing the effect of the degree of polynomial and point of estimation on the errors in parameter estimates. These curves are applicable in the special cases under discussion. Note that our comparisons are for the special case $\gamma_\theta = \gamma_\phi = 0$. It is seen from Eq. (4.11), that if instead, $\gamma_r = 0$, $\gamma_\phi \neq 0$, $\gamma_\theta \neq 0$, the estimates of range and range rate are not effected at all by the β estimate. In this sense, our comparisons are for the worst case.

5. COMPUTATIONAL ASPECTS

The two basic algorithms were given in Sec. 3. For completeness, we give the corresponding flow diagrams in Figs. 5.1 and 5.2. It is seen that Fig. 5.1 is merely Fig. 1.1 expressed in terms of mathematical symbols instead of verbal descriptions. The coordinate conversions indicated in Fig. 1.1 are not explicitly shown as the $\underline{w}(t)$ coordinate system is based on radar coordinates and multiplication by the matrix H performs the necessary coordinate conversions.

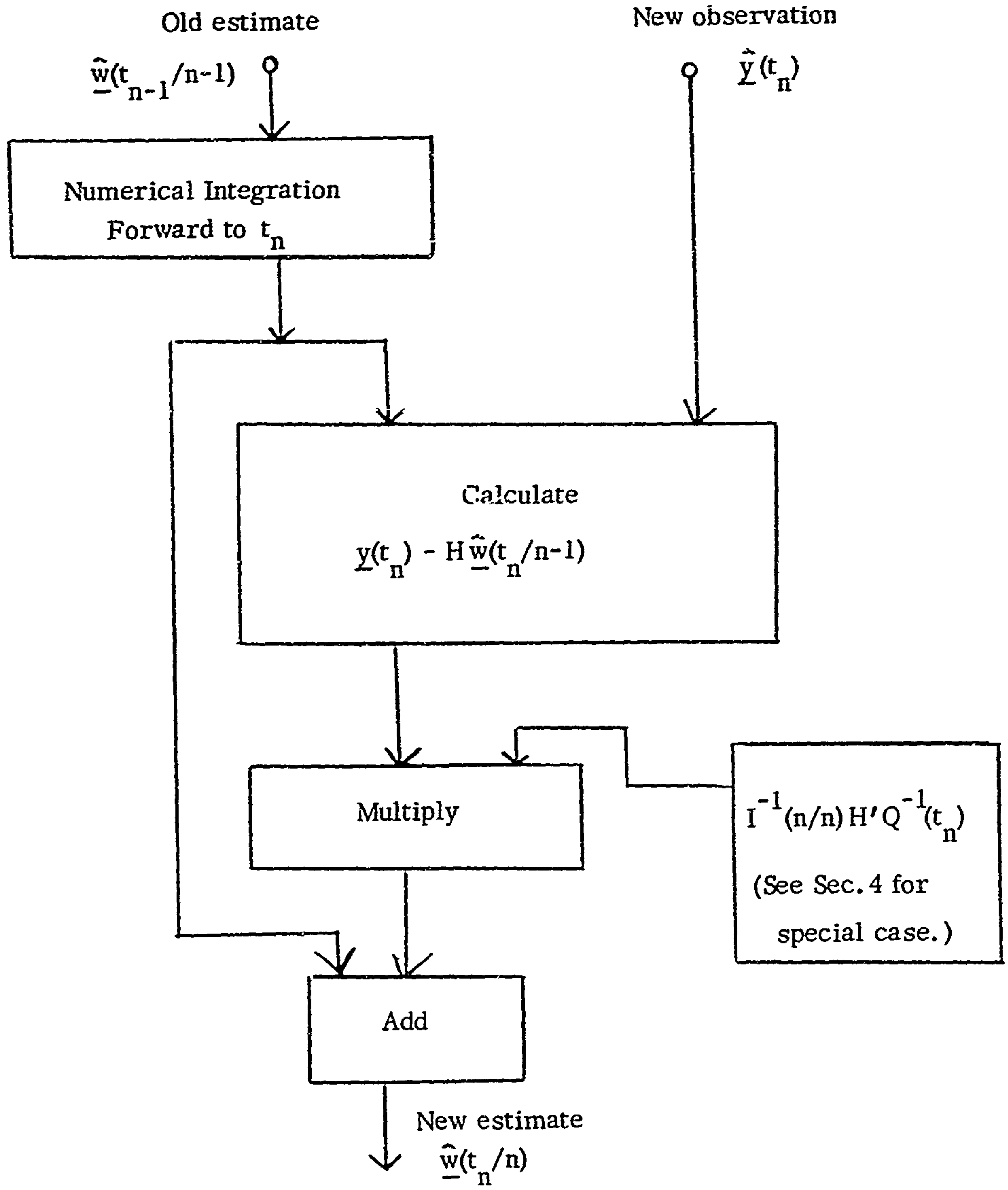

Fig. 5.1 Flow diagram for Eqs. (3.2) - (3.4) .

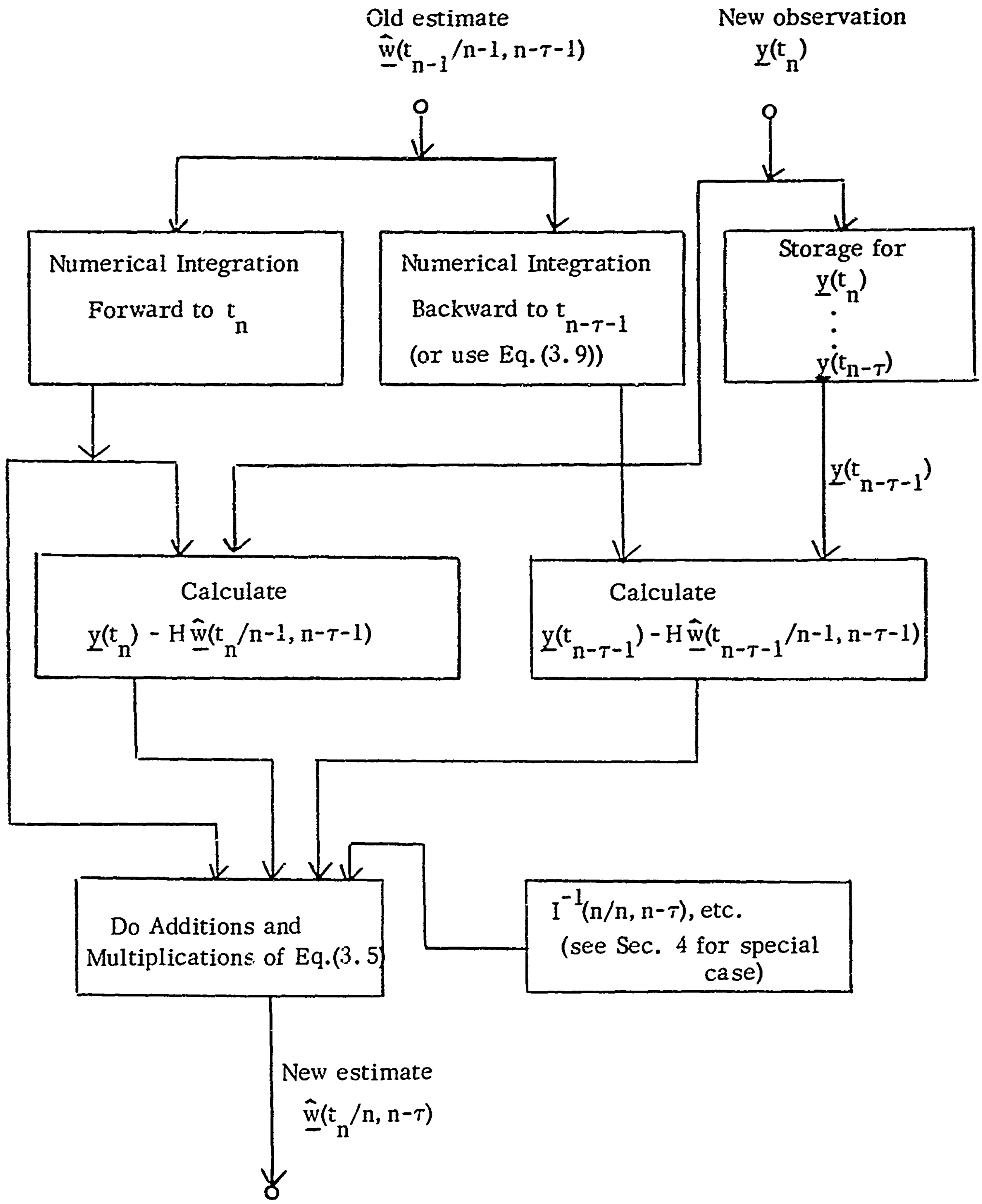

Fig. 5.2 Flow diagram for Eqs. (3.5) - (3.8) .

These algorithms are not quite ready for the programmer's pencil in that certain questions of implementation can only be resolved using the particulars of the actual problem. We now give a conglomerate of discussions relating to such mundane, but very important questions.

We have presented two different algorithms. The algorithm of Eqs. (3.2) through (3.4) is the easiest to implement and requires the least amount of computation time, but it is basically an ad hoc procedure. The algorithm of Eqs. (3.5) through (3.9) uses a chosen finite memory span but this advantage is obtained at the expense of a more complicated algorithm which requires some storage and an increased computation time which may or may not be important. Obviously the choice between the two algorithms depends on the problem of interest. However, for "scientific" post flight analyses, the preciseness of Eqs. (3.5) - (3.9) appears to override any computational disadvantages.

The parameters n_o and τ determine the effective memory span of the algorithm. Let T_1 be the maximum time interval over which β is essentially constant and Eq. (2.3) is a good approximation. Let T_2 be the maximum time interval over which the approximations of Sec. 4 are valid. Let $T = T_1$, or $T = \min \{T_1, T_2\}$ depending on whether or not the equations of Sec. 4 are used. It might be considered reasonable to choose memory spans corresponding to T. However, the problem can rarely be so easily resolved. First of all, T is difficult to specify as models such as a constant β or approximations *a la* Sec. 4 almost always fail gradually; that is, a small increase in T means the model and approximations are only slightly worse. Secondly, even if T were specified, a longer memory span might still be desirable to increase the ability of the algorithm to smooth out the effects of measurement noise. Thus compromises between the effects of model and approximation errors and the effect of the measurement errors might be required. A third complication is the presently unknown relationship between n_o and the effective memory span of the algorithm of Eqs. (3.2) - (3.4). A final difficulty is the altitude and geometry dependence of T. Thus even for a given re-entry, no single value of n_o or τ need be optimum for the entire flight. The overall outlook, however, is not really so dismal as the memory span is an easily

varied parameter and good values can be learned through experience. A preprogrammed memory span dependence on altitude and slant range is easy to provide. Furthermore, in many instances the ultimate in accuracy is not needed and overly conservative (i.e. small) memory spans are acceptable. If the ultimate is desired for post flight analysis, a man can monitor the results of repeated processing to see which memory span works best and make the appropriate adjustments.

The matrix, $Q(t_n)$, is fixed by the observation error distributions. Time variations in Q may arise in various ways; for example, when the signal-to-noise ratio is used to measure the variances of the observation errors. Note, however, that it is the relative magnitude of the various elements of Q that is important, not their absolute value (unless $\mathcal{I}^{-1}(n/n)$ is to be associated with the estimate errors). For many applications, a constant Q will be satisfactory.

The algorithms are based on the assumption of white observation errors; that is, statistical independence in time. This is often not true as, for example, range and angle tracking loops may be incorporated within the radar itself. If the memory span of the algorithm is sufficiently long compared to the error correlation time, little harm is done although $\mathcal{I}^{-1}(n/n)$ is then no longer the covariance matrix of the errors.* However, it is often advantageous to simply feed the algorithm with observations taken far enough apart in time to be "effectively" uncorrelated. If these time intervals are different for different observation coordinates such as range and angle, the ideas behind Eq. (3.10) can be used. Alternately, the values of Q can be modified to correspond to an "equivalent" white noise process.

Our algorithms estimate the body's state, $\underline{w}(t)$, at time t_n given the observations up to time t_n. For post flight analysis,estimates at a time, t_m, in the middle of the memory span are of more interest. $\underline{w}(t_m/n)$ can be calculated from $\underline{w}(t_n/n)$ by numerical integration backwards from t_n to t_m. However, the algorithms can be modified to provide midpoint estimates directly.

* See, for example, Ref. 11 which discusses the asymptotic efficiency of least squares processing (i.e. a white noise filter) in the presence of correlated errors.

We have presented algorithms that estimate the re-entry body's position and velocity in terms of radar coordinates and $\alpha(t_n)$, i.e. the $\underline{w}(t_n)$ coordinate system. The choice of radar coordinates was made for several reasons. No coordinate conversions of the raw data are required. This is important in high data rate radars when the number of relinearizations is reduced using the ideas of Eq. (3.10). It also simplifies the incorporation of range rate and gives a simple form for Q. Finally, the residuals between the observations and the smoothed trajectory are automatically available and they are important in post flight analysis. However, these advantages might be overweighted when the approximations of Sec. 4 are used as Eq. (4.1) might be a better approximation in, say, an inertial coordinate system than in radar coordinates. If estimation in a nonradar coordinate system is desired, the algorithms can be appropriately modified by simply considering the coordinate converted radar observations as the actual observations. The evaluations of the corresponding $Q^{-1}(t_n)$ can be done using the well-known partial derivatives of the new coordinate system with respect to the radar coordinates.* If range rate observations are to be used, one approach is to do the coordinate conversions using an elevation and azimuth rate calculated from the past estimates and then remove their effect by assigning these "fake" observations infinite error variances.

We estimate $\alpha(t_n) = \frac{\rho(t_n)}{\beta}$ rather than β where $\rho(t_n)$ is the air density. This is valuable at high altitudes where $\rho(t) \to 0$ as it keeps various "gains" from "blowing up." It has another advantage in that a simple exponential atmosphere may prove satisfactory for the actual processing. For example, if we use the model

* The Kwajalein algorithm estimates position and velocity in an x, y, z system without appropriate modification of the Q matrix. However, since range rate is not used and β is not estimated, it can be proved that this does not degrade the performance of the algorithm. Unfortunately a proof, although straightforward, is not available in the unclassified literature.

$$\rho(t_n) = \rho_o e^{-\frac{h(t_n)}{h_o}} \tag{5.1}$$

where ρ_o and h_o are scale factors and $h(t_n)$ is the height of the body above the Earth at time t_n, then

$$\alpha(t_n) = \alpha(t_{n-1}) e^{-\left[\frac{h(t_n) - h(t_{n-1})}{h_o}\right]} . \tag{5.2}$$

Because of the tracking nature of the algorithms, Eq. (5.2) need be valid only over the memory span of the algorithm and the value of ρ_o need not be known. A programmed variation of h_o with altitude could also be included. Of course to estimate β from $\hat{\alpha}(t_n)$, a standard atmosphere should be used. (This use of an exponential model is especially fruitful in some real time applications where β itself is just a nuisance parameter.) However, the decision to estimate $\alpha(t_n)$ is not sacred and can be modified to estimate β or some other function of β if the need arises. For example, a mach number dependence might be incorporated.

We discussed the evaluation of Eq. (2.3) in Sec. 2 using numerical integration in an inertial coordinate system. However, in practice the choice of this coordinate system depends primarily on the programmer's whims. Furthermore, because of the algorithm's tracking action, the model for Eq. (2.3) need be valid only over the algorithm's memory span. Thus analytic approximations to these equations of motion may be acceptable in place of any numerical integration. This argument is analogous to that used for the exponential atmosphere.

The discussions of Sec. 4 are not the last definitive word on error analysis but they indicate that, in many applications, it will be satisfactory to use our algorithms outside the atmosphere. However, if desired, we can employ a preprogrammed

algorithm change to begin estimating β at some predetermined altitude. The position and velocity estimates obtained exo-atmospherically furnish the initial conditions for this re-entry phase. The algorithms for estimating just position and velocity exo-atmospherically are an obvious special case.

In Sec. 4, we presented an approximation to $\Theta[\varepsilon_n, t_m, \hat{\underline{w}}(t_m)]$. Although this approximation appears to have a wide range of application, situations may definitely occur where a more accurate evaluation is required. There are several ways to proceed. More accurate analytic approximations can be developed. Exact analytic albeit complex, formulae are probably possible assuming an exponential atmosphere. The most general approach however, is the use of numerical techniques and of the various possibilities, the following appears to be the most satisfactory. In concept, at least, we can write the 7-dimensional vector system of first-order, nonlinear differential equations

$$\frac{d}{dt}\underline{w}(t) = \underline{g}[\underline{w}(t), t]$$

where the elements of $\underline{g}$, $g_k[\underline{w}(t), t]$, $k = 1, \ldots, 7$ include dependence on the atmospheric density, earth rotation, etc. Define the seven by seven matrix

$$B[\underline{w}^o(t), t] = \begin{bmatrix} \dfrac{\partial g_1[\underline{w}(t), t]}{\partial w_1(t)} & \cdots & \dfrac{\partial g_1[\underline{w}(t), t]}{\partial w_7(t)} \\ \vdots & & \vdots \\ \dfrac{\partial g_7[\underline{w}(t), t]}{\partial w_1(t)} & \cdots & \dfrac{\partial g_7[\underline{w}(t), t]}{\partial w_7(t)} \end{bmatrix}$$

where the partial derivatives are evaluated for $\underline{w}^o(t)$ as calculated from $\hat{\underline{w}}(t_m)$. Then we have the matrix differential equation

$$\frac{d}{dt}\Theta[t, t_m, \underline{w}(t_m)] = B[\underline{w}^o(t), t]\ \Theta[t, t_m, \underline{w}(t_m)]$$

with the initial conditions

$$\Theta[t_m, t_m, \underline{w}(t_m)] = I$$

which can be solved by numerical integration from t_m to t_n.* $\Theta^{-1}[t_n, t_m, w(t_m)]$ can be found by numerical matrix inversion or by integrating backward from t_n to t_m.

For the general cases not covered by Sec. 4, the $\mathcal{I}(n/n)$ matrix can be numerically calculated by the recursive formula, Eqs. (3.4) or (3.6). Numerical matrix inversion is required to obtain $\mathcal{I}^{-1}(n/n)$. There are various ways to reduce the computation time required, see for example the discussions in Appendix A.7 of Ref. 4. The number of matrix inversions can be reduced by relinearizing only every r^{th} vector observation as illustrated by Eq. (3.10).

$\mathcal{I}(n/n)$ of Eq. (3.4) becomes constant for $n > n_o$. Similarly, in many cases, $\mathcal{I}(n/n, n-\tau)$ becomes essentially constant. In some applications, it might be acceptable to simply precalculate these constant values and just use them in the algorithms. This introduces additional transients into the system but they may not be important. Similarly, in the special case of Sec. 4 it might be satisfactory to merely use the formulae for the case where $n \gg 1$ which simplifies the expressions.

* In practice, the integration would probably be done in inertial coordinates and then converted into the $\underline{w}(t)$ system.

6. DISCUSSION

The algorithms of this report have not yet been programmed and used. However, the basic underlying ideas have been proven to be sound by analysis, simulation and field application. The approximation of Sec. 4 is physically reasonable and leads to an intuitively satisfying system. We therefore feel the algorithms are ready for implementation using the approximations of Sec. 4. Questions such as memory length and actual performance are best resolved by running the algorithms.

We have purposely addressed ourselves to a very special problem, namely the estimation of seven parameters, position, velocity and β from a single radar. However, it is obvious that the theory is actually applicable in a far more general context. By introducing some more coordinate conversions and allowing variations in the matrix H, the technique can be used equally well for multiple sites, each with different tracking abilities; for example, the combination of Baker-Nunn and radar data. In addition, many other parameters can be included in the model such as radar bias errors, station location errors and more complex parameterizations of the body's motion. These various extensions can be implemented directly using the numerical techniques of Sec. 5 to evaluate Θ and T, but by some diligence, it is probable that simplified analytic approximations such as those given in Sec. 4 can also be developed. In fact, extension to a parameterization of a time-varying β such as

$$\beta(t) = \sum_{j=1}^{p} \beta_j t^j$$

where the β_j are considered unknown parameters requires a fairly trivial analysis provided the orthonormal polynomials of Sec. 4 are employed. The same is true for the incorporation of coherent range acceleration and the estimation of lift forces. In addition, if good doppler is available, it presently appears reasonable to estimate

body oscillations by modeling β in terms of a sinusoid of unknown frequency.* The basic recursive schemes can also be generalized to handle n^{th} order Markovian as well as white observation noise.

A critical parameter of the data processing algorithms is the memory span as determined by n_0 or τ. We have discussed the use of a preprogrammed altitude dependence for varying these quantities and also the use of a man to determine the best values in post flight analysis by repeated processing. However, the most interesting and powerful approach would be to use the radar data itself to determine the memory span. The basic tenets of such a technique are already known and are applicable not only to memory span control but also to problems such as an automatic transition from an exo-atmospheric to a re-entry algorithm. The analysis of the technique will be helped by the analogy, mentioned in Sec. 3, between the basic recursive algorithms and feedback control theory. As with most nonlinear control systems, simulation on a computer complex capable of man-machine interaction will probably be required. This may be an area of future research.

* Recursive techniques similar to those of this report exist for estimating an unknown frequency. A Lincoln Laboratory report, "Computationally Feasible Frequency Estimates in the Presence of Unknown Phase and Amplitude" by L. A. Gardner, Jr., is presently in preparation.

REFERENCES

1. F. C. Schweppe, "Reduction of Computational and Data Transmission Requirements for Trajectory Estimation Using Multiple Sites," 22G-10, Lincoln Laboratory, M. I. T., (23 August 1963).

2. L. A. Gardner, Jr., "Recursive Estimation Schemes for Certain Nonlinear Regression Problems," JA-2448, Lincoln Laboratory, M. I. T., September 4, 1964, to be published.

3. A. Albert, L. A. Gardner, Jr., "Stochastic Approximation and Nonlinear Regression," JA-2430, Lincoln Laboratory, M. I. T., August 20, 1964.

4. F. C. Schweppe, "An Introduction to Estimation Theory for Dynamical Systems," 22G-15, Lincoln Laboratory, M. I. T., (August 28, 1963).

5. P. Swerling, "First-Order Error Propagation in a Stagewise Smoothing Procedure for Satellite Observations," Rand Corporation, RM-2329, (June 15, 1959).

6. G. Smith, "Multi-variable Linear Filter Theory Applied to Space Vehicle Guidance," J. SIAM Control, Ser. A, 12, No. 1, (1964).

7. S. Kullback, Information Theory and Statistics, (John Wiley and Sons, Inc., New York, N. Y.).

8. F. C. Schweppe, "Optimization of Signals," 1964-4, Lincoln Laboratory, M. I. T. (January 1964).

9. F. C. Schweppe, "On the Linear Smoothing of Redundant Radar Data From Satellites," 22G-0021, Lincoln Laboratory, M. I. T., (August 25, 1960).

10. R. Crittendon, E. Dubinsky, D. Lubell, F. Schweppe, "A Number of Results from Studies of Digital Filtering and the Convergence of Iterative Techniques," 22G-0061, Lincoln Laboratory, M. I. T., (June 1, 1961).

11. U. Grenander, M. Rosenblatt, Statistical Analysis of Stationary Time Series, (John Wiley and Sons, New York, N. Y.)

DISTRIBUTION LIST

Division 2

F.C. Frick
S.H. Dodd

Group 21

H.L. Kasnitz
G.M. Shannon
C.W. Uskavitch

Group 22

V.A. Nedzel
W.Z. Lemnios
W.I. Wells
A. Freed
A.A. Grometstein
J.H. Hartogensis
C.E. Nielsen, Jr.
J. Rheinstein
A.F. Smith
P.A. Willmann

Group 28

J.A. Arnow
J.F. Nolan
A.W. Armenti
C.R. Arnold
M. Athans
T.C. Bartee
R.N. Davis
P.L. Falb
L.A. Gardner, Jr.
P.L. Grant
H.K. Knudsen
J.B. Lewis
H.E. Meily
A.J. Morency
H.C. Peterson
F.C. Schweppe (20)
J.M. Winett
P.E. Wood, Jr.

Division 3

W.R. Romaine

Group 35

L.R. Martin
J. Penhune
K.E. Ralston
A.R. Warwas

Division 4

J. Freedman
H.G. Weiss

Group 41

M. Axelbank
H.M. Jones
H.O. Schneider
J.P. Hawkes

Group 42

D.A. Cahlander

Group 45

W.W. Ward
J.P. Perry
A. Bertolini
S.F. Catalano
D.F. DeLong
H.W. Eklund
R.W. Straw

Group 62

F.E. Heart

WSMR

S.J. Miller
G.R. Armstrong
R.L. Hardy
C.J. McKenney

Kwajalein

T.H. Einstein
E. Korngold
B. Nanni
R.P. Wishner

Hq. ESD
(Maj. S. Jeffery ESRC)
L.G. Hanscom Field
Bedford, Mass.

Hq. ESD (Lt. J. Egan)
L.G. Hanscom Field
Bedford, Mass.

FCS: smm/jmk
3 December 1964